Как правильно выбирать очередь

Владимир Перепелица



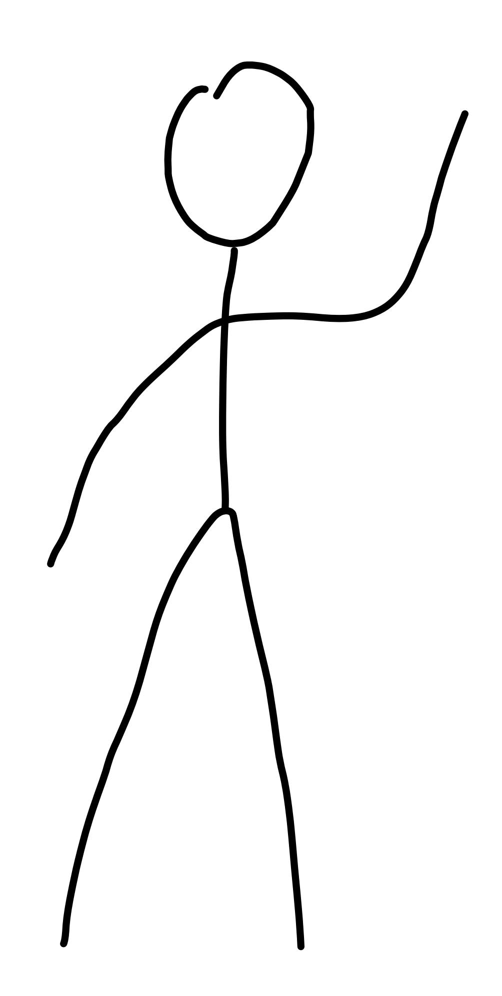
Как правильно выбирать очередь

Mons Anderson



Кто я?

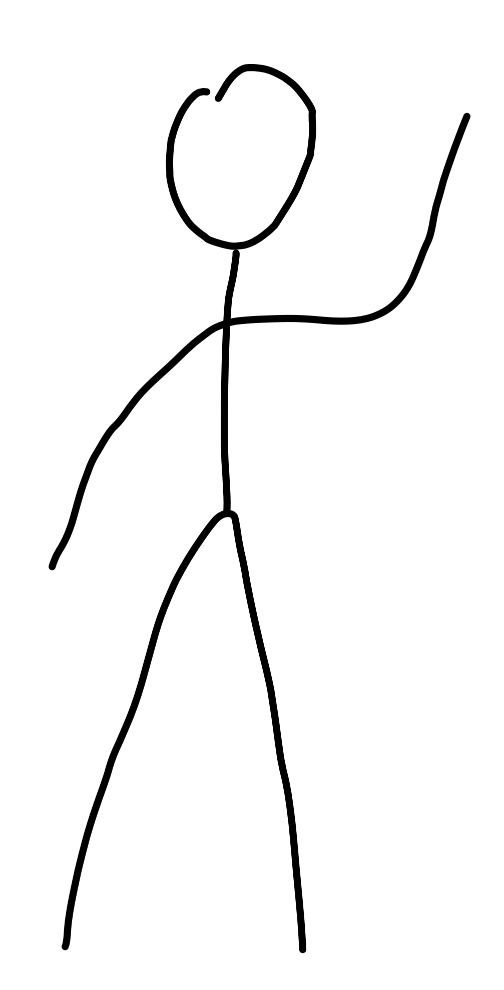
- Архитектор Облака Mail.ru, Mail.ru Cloud Solutions
- Архитектор и продакт-менеджер Tarantool





Кто я?

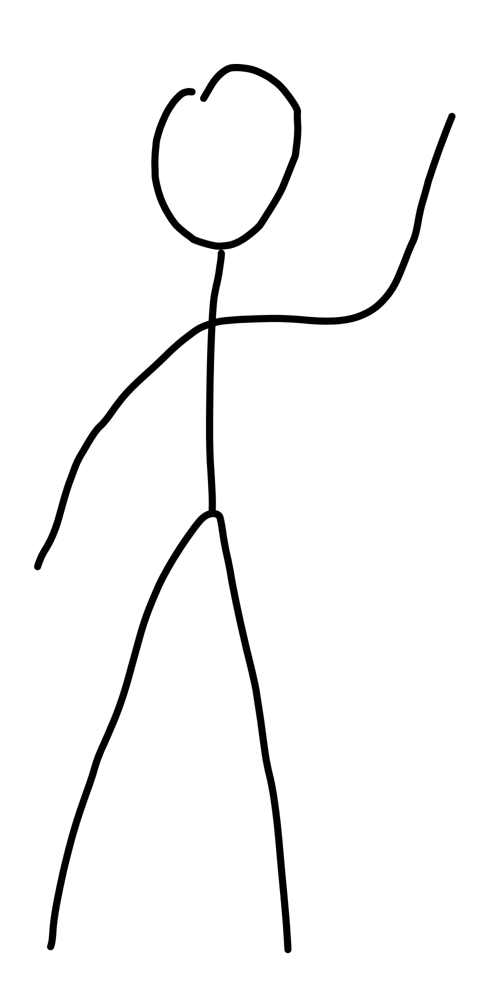
- Архитектор Облака Mail.ru, Mail.ru Cloud Solutions
- Архитектор и продакт-менеджер Tarantool
- Использую очереди с 2008 года





Кто я?

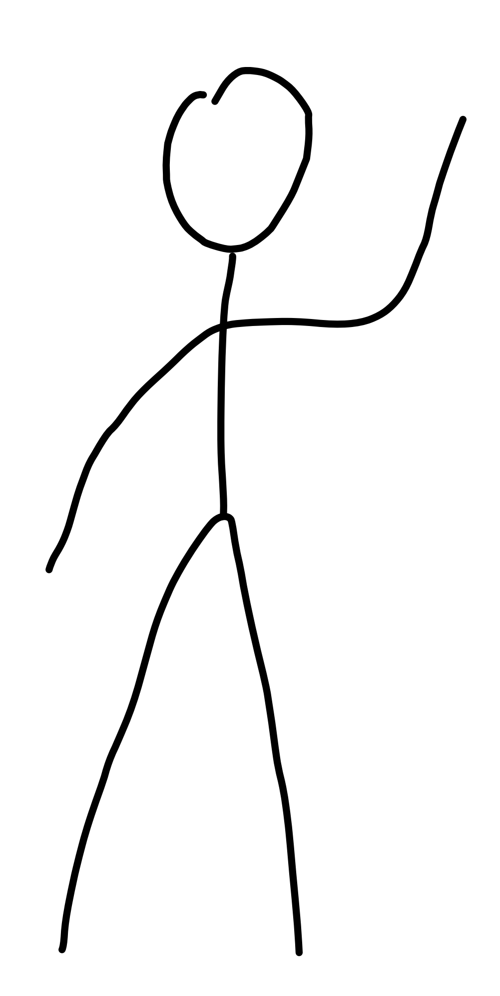
- Архитектор Облака Mail.ru, Mail.ru Cloud Solutions
- Архитектор и продакт-менеджер Tarantool
- Использую очереди с 2008 года
- Люблю реализовывать очереди





Кто я?

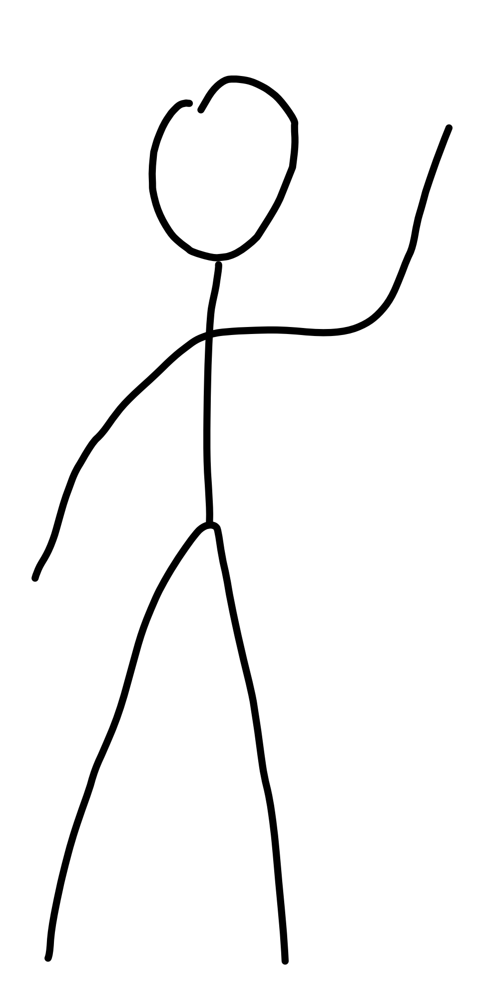
- Архитектор Облака Mail.ru, Mail.ru Cloud Solutions
- Архитектор и продакт-менеджер Tarantool
- Использую очереди с 2008 года
- Люблю реализовывать очереди (на **Tarantool** :)



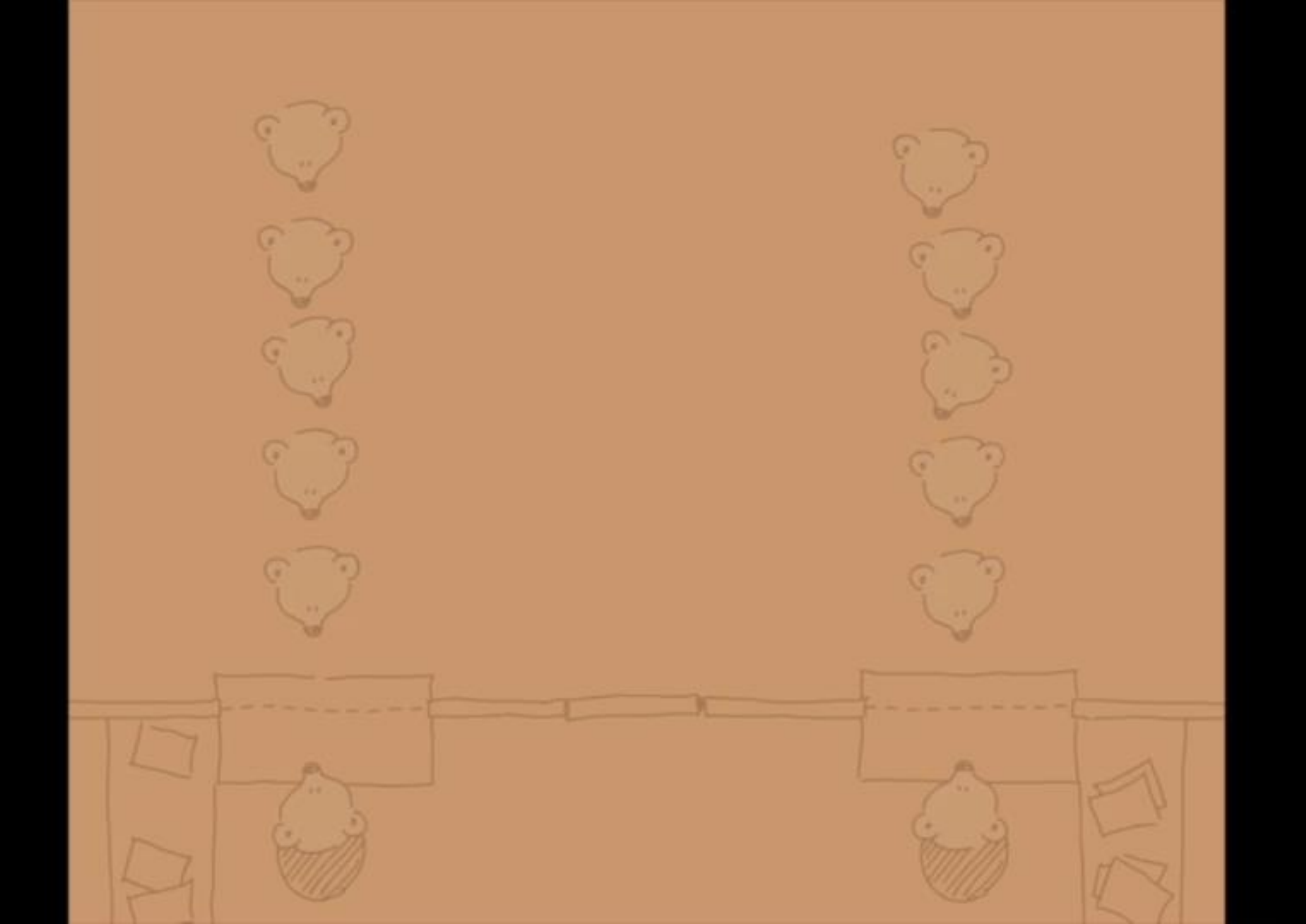


Кто я?

- Архитектор Облака Mail.ru, Mail.ru Cloud Solutions
- Архитектор и продакт-менеджер Tarantool
- Использую очереди с 2008 года
- Люблю реализовывать очереди (на Tarantool)
- Люблю рассказывать про очереди





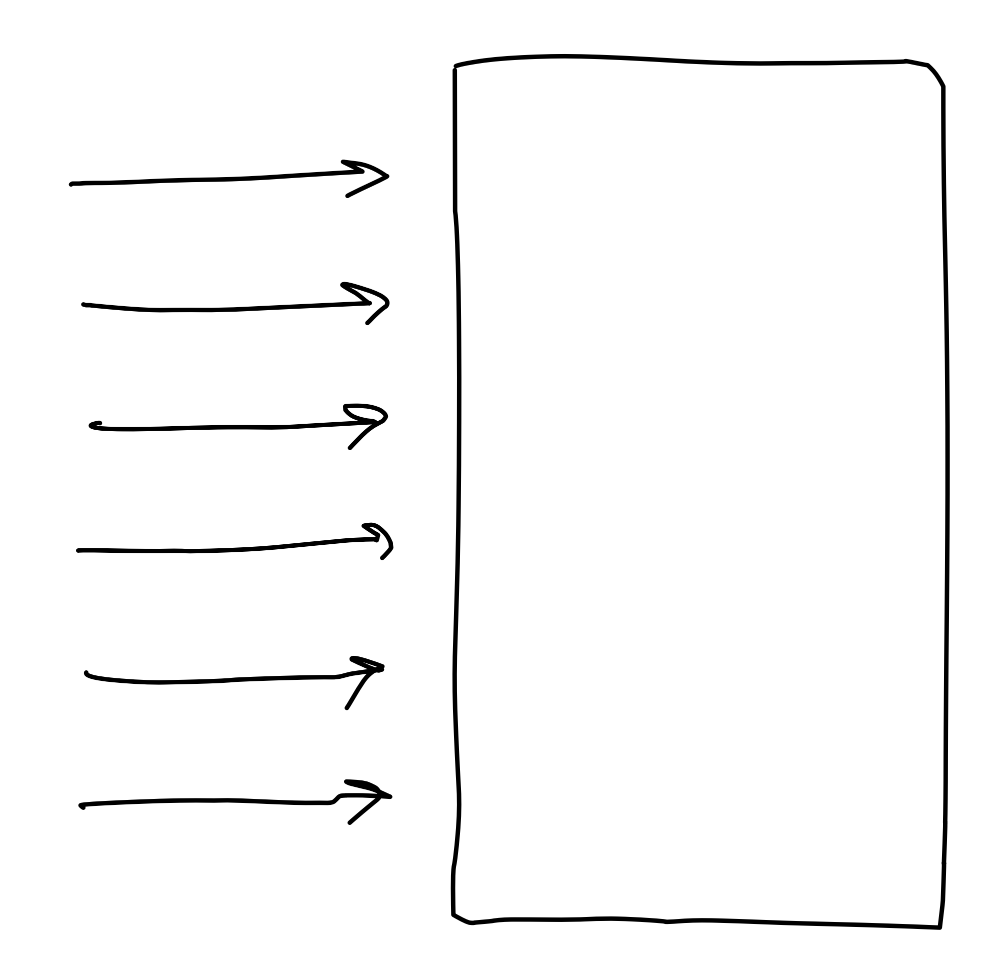


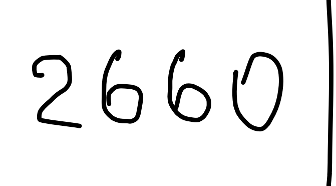
Зачем нужны очереди?

- Распределение задач

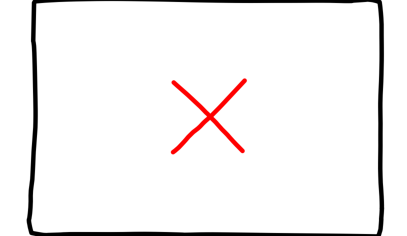


Распределение задач





6230





Распределение задач

N

2660

6230



Распределение задач

2660

6230



Зачем нужны очереди?

- Распределение задач
- Планирование исполнения



Планирование исполнения

RPS

Пик нагрузки

t



Планирование исполнения

RPS

Оптимум

Очередь

t



Зачем нужны очереди?

- Распределение задач
- Планирование исполнения
- Честность выделения ресурсов



Зачем нужны очереди?

- Распределение задач
- Планирование исполнения
- Честность выделения ресурсов
- Репликация сообщений



Зачем нужны очереди?

- Распределение задач
- Планирование исполнения
- Честность выделения ресурсов
- Репликация сообщений
- Отказоустойчивость, надёжность, гарантия доставки



Зачем нужны очереди?

- Распределение задач
- Планирование исполнения
- Честность выделения ресурсов
- Репликация сообщений
- Отказоустойчивость, надёжность, гарантия доставки
- Коммуникация микросервисов



Зачем нужны очереди?

- Распределение задач
- Планирование исполнения
- Честность выделения ресурсов
- Репликация сообщений
- Отказоустойчивость, надёжность, гарантия доставки
- Коммуникация микросервисов
- Событийная архитектура (Event Sourcing)



Зачем нужны очереди?

- Распределение задач
- Планирование исполнения
- Честность выделения ресурсов
- Репликация сообщений
- Отказоустойчивость, надёжность, гарантия доставки
- Коммуникация микросервисов
- Событийная архитектура (Event Sourcing)
- Потоковая архитектура (Streaming)



Где применяются очереди?

- «Железо»

IRQ
NCQ
Hardware Buffers



Где применяются очереди?

- «Железо»
- Ядро операционной системы

epoll / kqueue
networking
signal handling



Где применяются очереди?

- «Железо»
- Ядро операционной системы
- Приложения

Cross thread
IPC



Где применяются очереди?

- «Железо»
- Ядро операционной системы
- Приложения
- Сетевые взаимодействия



Где применяются очереди?

- «Железо»
- Ядро операционной системы
- Приложения
- Сетевые взаимодействия
- Распределённые системы



Где применяются очереди?

- «Железо»
- Ядро операционной системы
- Приложения
- Сетевые взаимодействия
- Распределённые системы
- Стык разных бизнесов



Где применяются очереди?

- «Железо»
- Ядро операционной системы
- Приложения
- Сетевые взаимодействия
- Распределённые системы
- Стык разных бизнесов

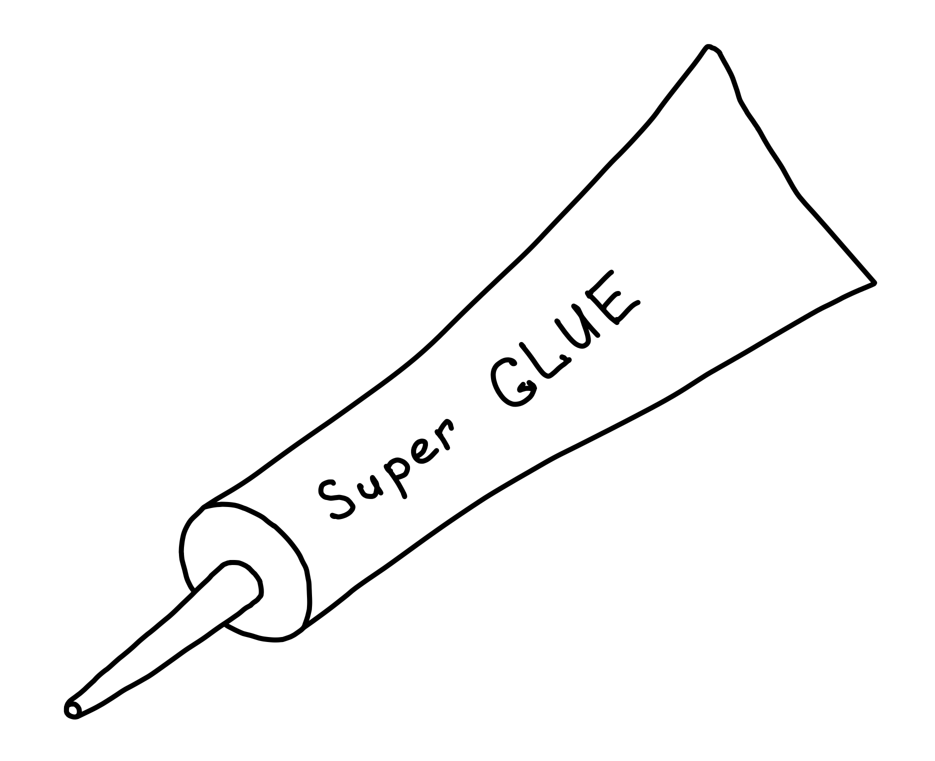
- Фактически — везде



Где применяются очереди?

- «Железо»
- Ядро операционной системы
- Приложения
- Сетевые взаимодействия
- Распределённые системы
- Стык разных бизнесов

- Фактически — везде



Очереди — это клей



Что такое очередь?

- Средство коммуникации при помощи сообщений



Что такое очередь?

m

enqueue

m m m m

dequeue



Что такое очередь?

- Средство коммуникации при помощи сообщений
- Подход Put/Take: 1 → 1



Подход Put/Take

Put

Take

Release

Ack



Что такое очередь?

- Средство коммуникации при помощи сообщений
- Подход Put/Take: 1 → 1
- Подход Publish/Subscribe: 1 → *



Подход Pub/Sub

Publish

Subscribe

topic

topic

P

P

P

C

C

C



Что такое очередь?

- Средство коммуникации при помощи сообщений
- Подход Put/Take: 1 → 1
- Подход Publish/Subscribe: 1 → *
- Подход Request/Response: 1 ⇆ 1



Что такое очередь?

- Средство коммуникации при помощи сообщений
- Подход Put/Take: 1 → 1
- Подход Publish/Subscribe: 1 → *
- Подход Request/Response: 1 ⇆ 1
- Протоколы: AMQP, MQTT, STOMP, NATS, ZeroMQ, …



Какие есть варианты?

- Облачные решения
  - Amazon **SQS**
  - Mail.ru Cloud Queues
  - Yandex Message Queue
  - CloudAMQP
  - …



Какие есть варианты?

- Облачные решения
  - Amazon **SQS**, Mail.ru Cloud Queues, Yandex MQ, CloudAMQP, …
- Специализированные брокеры
  - **RabbitMQ**
  - Apache **Kafka**
  - ActiveMQ
  - Tarantool Queue
  - **NATS**
  - NSQ
  - …



Какие есть варианты?

- Облачные решения
  - Amazon **SQS**, Mail.ru Cloud Queues, Yandex MQ, CloudAMQP, …
- Специализированные брокеры
  - **RabbitMQ**, Apache **Kafka**, ActiveMQ, Tarantool Queue, **NATS**, NSQ, Beanstalkd, …
- Реализация очереди с помощью СУБД
  - PgQueue
  - Tarantool
  - Redis
  - …



Какие есть варианты?

- Облачные решения
  - Amazon **SQS**, Mail.ru Cloud Queues, Yandex MQ, CloudAMQP, …
- Специализированные брокеры
  - **RabbitMQ**, Apache **Kafka**, ActiveMQ, Tarantool Queue, **NATS**, NSQ, Beanstalkd, …
- Реализация очереди с помощью СУБД
  - PgQueue, Tarantool, Redis, …
- «Сокеты на стероидах»
  - NATS, ZeroMQ



Основные кандидаты

- Apache Kafka
  - Распределённый лог сообщений для стриминга



Основные кандидаты

- Apache Kafka
  - Распределённый лог сообщений для стриминга
- RabbitMQ
  - Традиционный брокер с протоколом AMQP



Основные кандидаты

- Apache Kafka
  - Распределённый лог сообщений для стриминга
- RabbitMQ
  - Традиционный брокер с протоколом AMQP
- Managed Cloud Queue (SQS/MQ/…)
  - Максимальное удобство в облаках



Основные кандидаты

- Apache Kafka
  - Распределённый лог сообщений для стриминга
- RabbitMQ
  - Традиционный брокер с протоколом AMQP
- Managed Cloud Queue (SQS/MQ/…)
  - Максимальное удобство в облаках
- NATS
  - Связующее звено для микросервисов

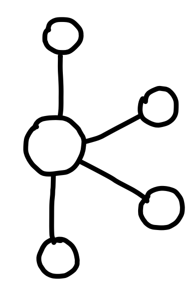


Основные кандидаты

- Apache Kafka
  - Распределённый лог сообщений для стриминга
- RabbitMQ
  - Традиционный брокер с протоколом AMQP
- Managed Cloud Queue (SQS/MQ/…)
  - Максимальное удобство в облаках
- NATS
  - Связующее звено для микросервисов
- Tarantool
  - Платформа для произвольных очередей



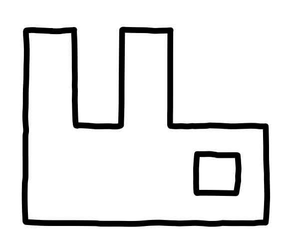
Apache Kafka



- Реплицированный шардированный лог сообщений
- Строгая упорядоченность (FIFO)
- Ограничена по количеству потребителей
- Повторное проигрывание последовательности
- Интеграция с экосистемой Apache
- Основные сценарии использования
  - Анализ данных. Логи, метрики, аудит
  - Производительный процессинг потоковых данных
  - Репликация данных



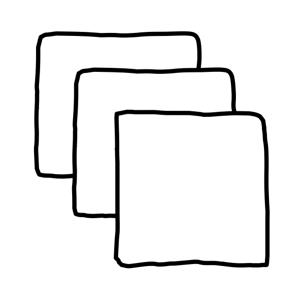
RabbitMQ



- Протоколы: AMQP, MQTT, STOMP
- Приоритеты, отложенные и фоновые задачи
- Нет ограничений на количество потребителей
- Хранение: память, диск, репликация, кворум
- Простой в освоении. Сложный в отказоустойчивости
- Основные сценарии использования
  - Традиционный pub/sub брокер
  - Слой соединения микросервисов. Шина сообщений



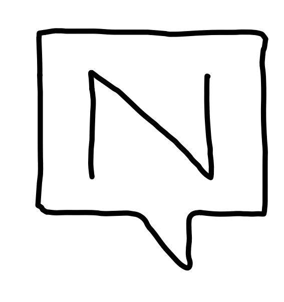
Managed Cloud Queue



- Надёжная, автоматически масштабируемая очередь
- Протокол без состояния: простая коммуникация
- Стандартизированное API
- Минимум затрат при низком потреблении
- Основные сценарии использования
  - Коммуникация между сервисами в облаке
  - Связующее звено для S3 и Lambda



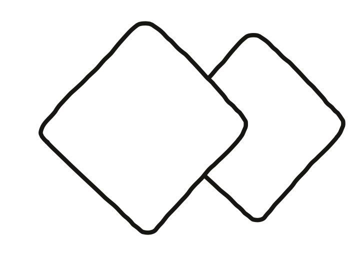
NATS Messaging



- Быстрый неперсистентный обмен сообщениями
- Высокая производительность и масштабируемость
- Любые сценарии: pub/sub, put/take, req/res
- При использовании JetStream
  - Потоковая обработка
  - Надёжное хранение, RAFT cluster
- Основные сценарии использования
  - Инструмент для общения в распределённых системах



Tarantool

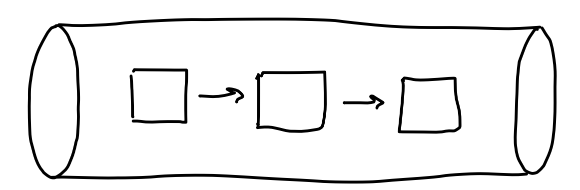


- Готовый брокер с репликацией (Tarantool Queue)
- Интеграция со стриминговыми очередями
- Модули для построения собственных очередей
- Любая произвольная логика
- Транзакционность в рамках одного брокера
- Основные сценарии использования
  - Производительный брокер для традиционных сценариев
  - Построение сложных очередей с собственной логикой



Проблемы очередей: алгоритмы

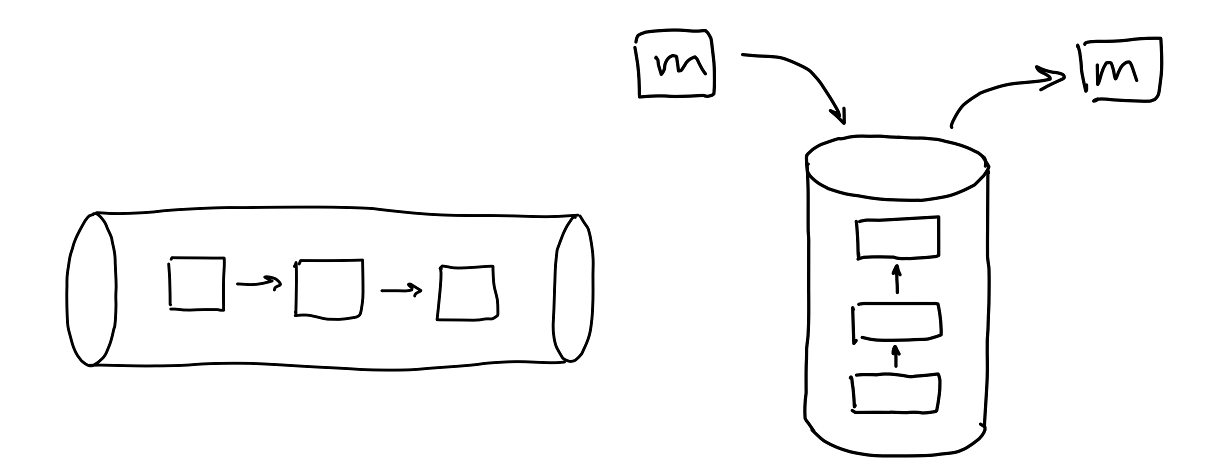
- FIFO, LIFO, Best Effort, QoS





Проблемы очередей: алгоритмы

- FIFO, LIFO, Best Effort, QoS





Проблемы очередей: алгоритмы

- FIFO, LIFO, Best Effort, QoS



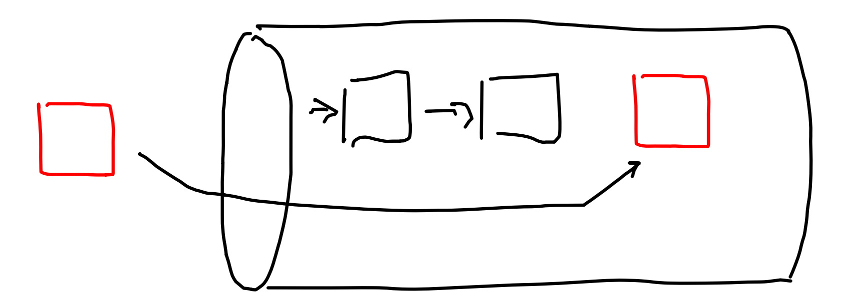
Проблемы очередей: алгоритмы

- FIFO, LIFO, Best Effort, QoS



Проблемы очередей: алгоритмы

- FIFO, LIFO, Best Effort, QoS
- Приоритизация сообщений





Проблемы очередей: алгоритмы

- FIFO, LIFO, Best Effort, QoS
- Приоритизация сообщений
- Организация подочередей



Проблемы очередей: алгоритмы

- FIFO, LIFO, Best Effort, QoS
- Приоритизация сообщений
- Организация подочередей
- Повтор, отложенные задачи, повтор с задержкой



Проблемы очередей: алгоритмы

- FIFO, LIFO, Best Effort, QoS
- Приоритизация сообщений
- Организация подочередей
- Повтор, отложенные задачи, повтор с задержкой
- Dead letter queue (и упорядочивание)



Проблемы очередей: алгоритмы

- FIFO, LIFO, Best Effort, QoS
- Приоритизация сообщений
- Организация подочередей
- Повтор, отложенные задачи, повтор с задержкой
- Dead letter queue (и упорядочивание)
- Созависимые задачи



Проблемы очередей: алгоритмы

- FIFO, LIFO, Best Effort, QoS
- Приоритизация сообщений
- Организация подочередей
- Повтор, отложенные задачи, повтор с задержкой
- Dead letter queue (и упорядочивание)
- Созависимые задачи
- TTL, TTR, Putback



Проблемы очередей: алгоритмы

- Приоритизация и голодание (*Starvation*)



Проблемы очередей: алгоритмы

- Приоритизация и голодание (*Starvation*)
- Пропускная способность (*Throughput*)



Проблемы очередей: алгоритмы

- Приоритизация и голодание (*Starvation*)
- Пропускная способность (*Throughput*)
- Производительность (*Performance*)



Проблемы очередей: алгоритмы

- Приоритизация и голодание (*Starvation*)
- Пропускная способность (*Throughput*)
- Производительность (*Performance*)
- Масштабируемость (*Scalability*)



Проблемы очередей: алгоритмы

- Приоритизация и голодание (*Starvation*)
- Пропускная способность (*Throughput*)
- Производительность (*Performance*)
- Масштабируемость (*Scalability*)
- Ограниченность (*Capacity*)



Проблемы очередей: алгоритмы

- Приоритизация и голодание (*Starvation*)
- Пропускная способность (*Throughput*)
- Производительность (*Performance*)
- Масштабируемость (*Scalability*)
- Ограниченность (*Capacity*)
- Сохранность сообщений (*Durability*)*



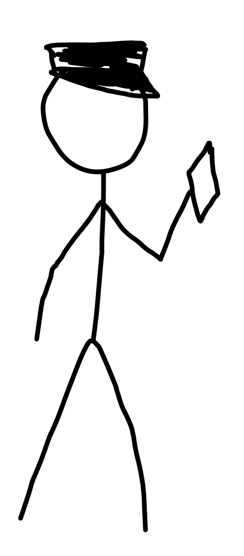
Проблемы очередей: сеть

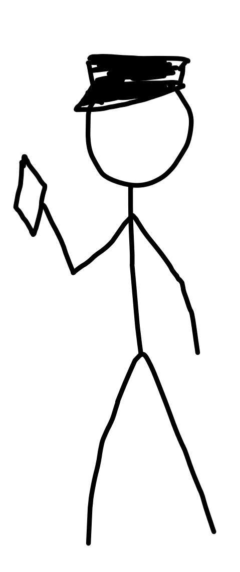
- Undefined behavior



Проблемы очередей: сеть

- Проблема двух генералов

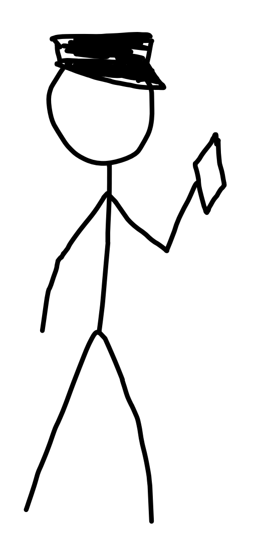
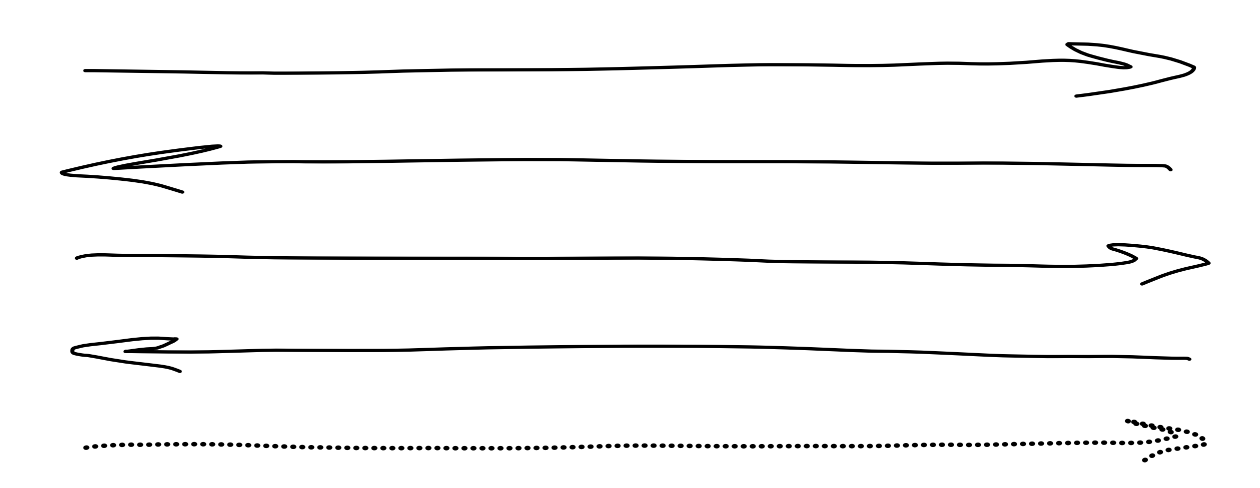
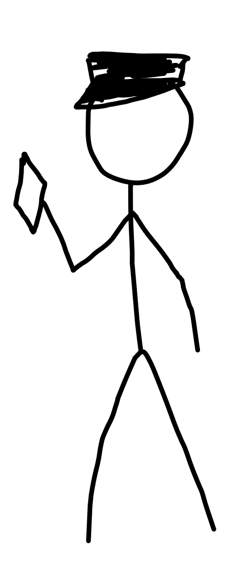






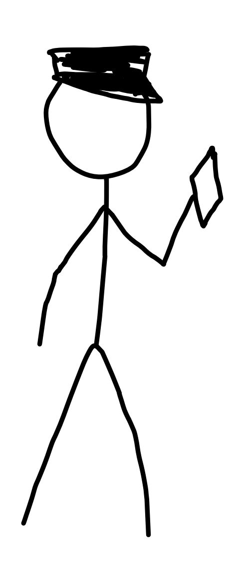
Проблемы очередей: сеть

- Проблема двух генералов


Проблемы очередей: сеть

- Проблема двух генералов



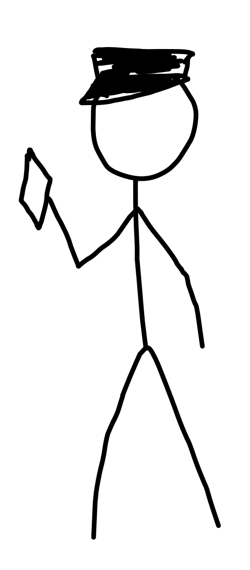
Есть две сложные проблемы

в распределённых системах:

2. Доставка строго один раз

1. Строгий порядок сообщений

2. Доставка строго один раз





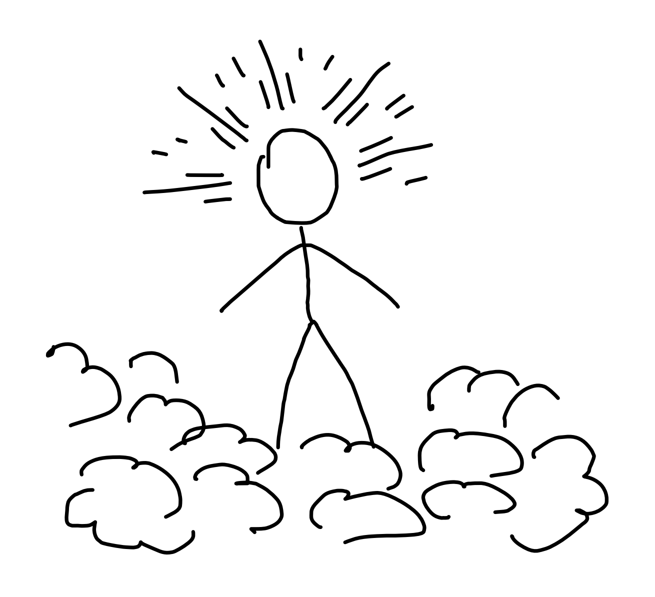
Exactly once

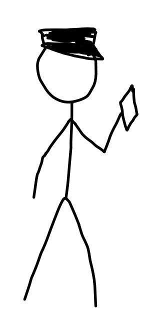


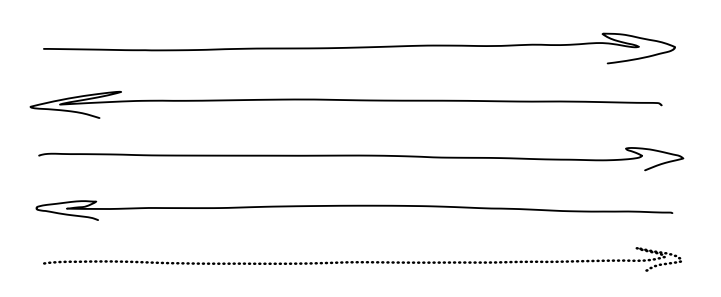
Exactly once

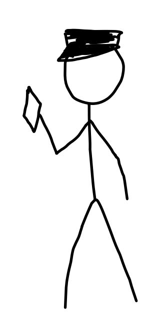


«Решение» проблемы «exactly once»











Проблемы очередей: сеть и диск

- Пропускная способность (*Throughput*)
- Задержка в обработке (*Latency*)



Проблемы очередей: отказы

- Оборудование
  - Диск
  - Хост
  - Дата-центр



Проблемы очередей: отказы

- Оборудование
  - Диск
  - Хост
  - Дата-центр
- Временный отказ
  - Питание
  - Сеть
  - Split brain



Проблемы очередей: отказы

- Оборудование
  - Диск
  - Хост
  - Дата-центр
- Временный отказ
  - Питание
  - Сеть
  - Split brain
- Отказ навсегда
  - Физическое уничтожение



Проблемы очередей: отказы

- Оборудование
  - Диск
  - Хост
  - Дата-центр
- Временный отказ
  - Питание
  - Сеть
  - Split brain
- Отказ навсегда
  - Физическое уничтожение

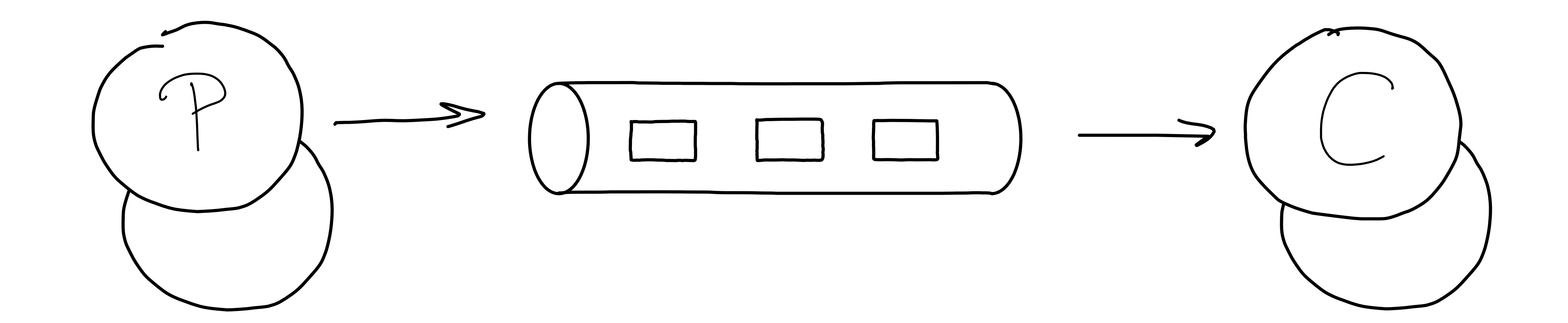


Проблемы очередей: отказы

- Доступность (*Availability*)
  - Возможность сохранить сообщение
- Надёжность (*Durability*)
  - Гарантия сохранности и доставки сообщения

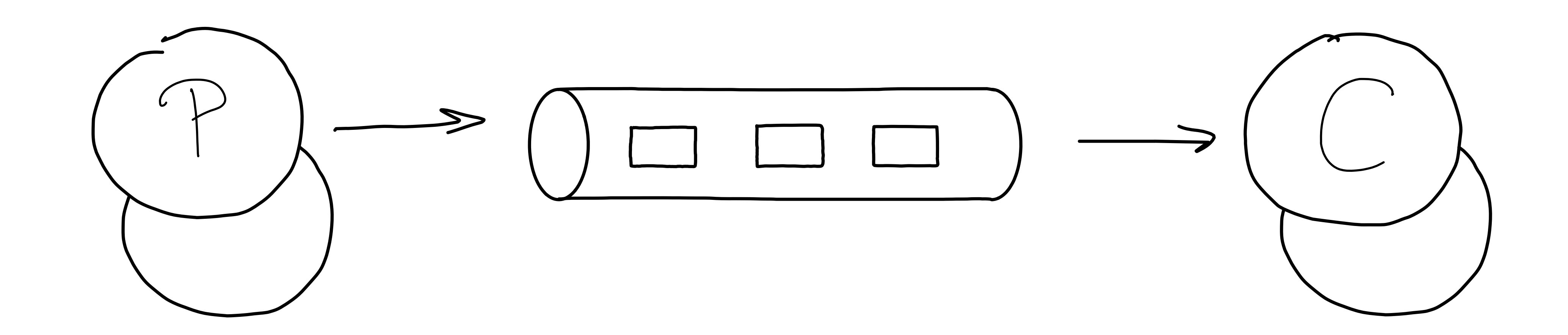


Топологии очередей: single instance





Топологии очередей: single instance

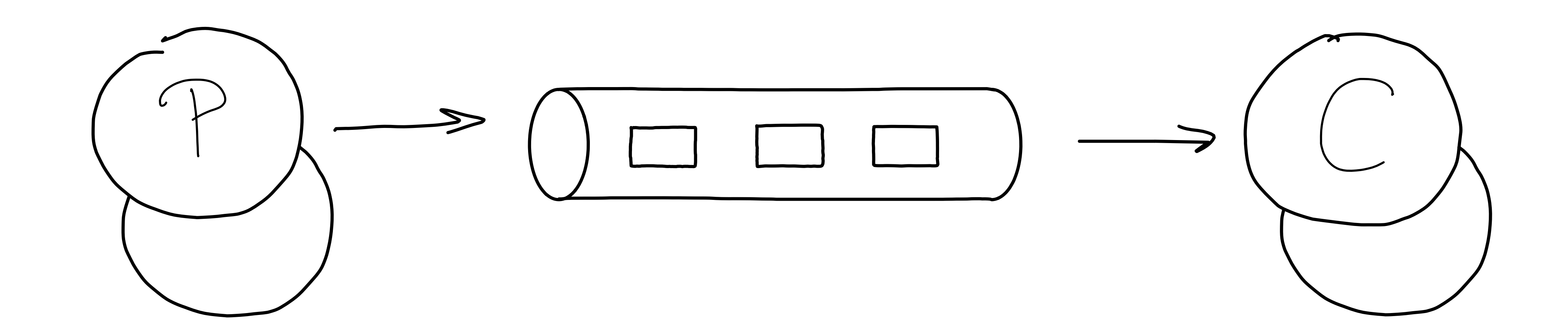


Масштабируемость: нет

Гарантии: $X \leq 1$, $X \geq 1$



Топологии очередей: single instance



Масштабируемость: нет

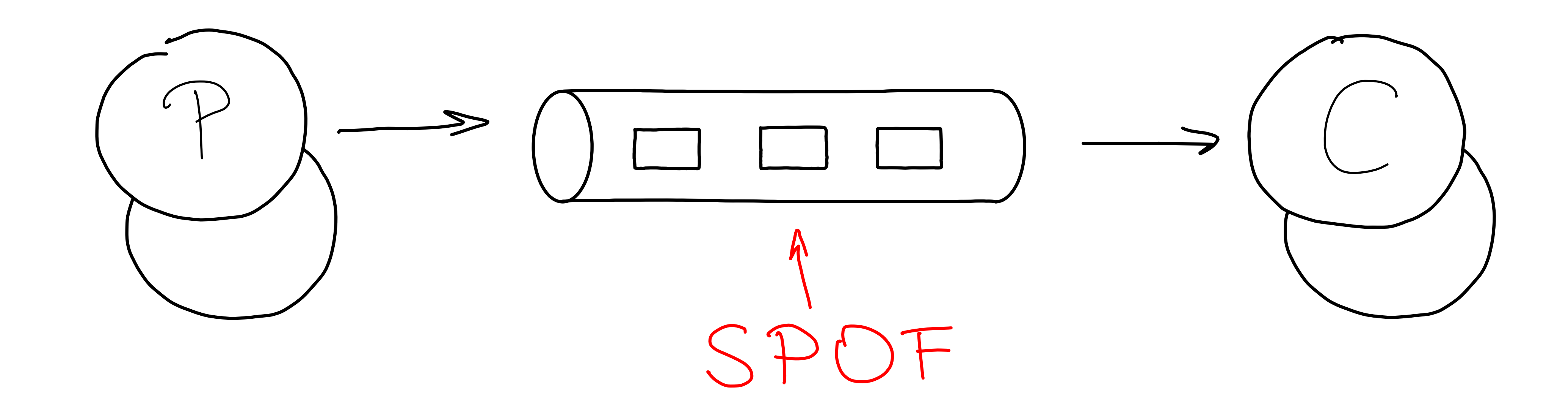
Гарантии: X ≤ 1, X ≥ 1

Доступность: низкая

Надёжность: низкая



Топологии очередей: single instance



Масштабируемость: нет

Гарантии: X ≤ 1, X ≥ 1

Доступность: низкая

Надёжность: низкая



Топологии очередей: multi-instance

P

1 of N

C

C

Масштабируемость: да

Гарантии: X ≤ 1, X ≥ 1



Несколько очередей, кладём в 1

P

1 of N

C

C

Масштабируемость: да

Гарантии: X ≤ 1, X ≥ 1

Доступность: высокая

Надёжность: средняя



Несколько очередей, кладём в K из N

P

K of N

3 2 1

3 2 1

C

C



Несколько очередей, кладём в K из N

Доступность: высокая

Надёжность: высокая



Несколько очередей, кладём в K из N

Подход: «многократное дублирование»

P

K of N

3 2 1

3 2 1

C

C

Масштабируемость: да

Гарантии: X ≥ K

Доступность: высокая

Надёжность: высокая



Репликация

P

L

S

C

Коммуникация
с лидером

Реплики
в ожидании



Реплицированные очереди, 1 из N

Подход: «как база данных»

P

1 of N

C

C

Масштабируемость: да

Гарантии: X ≈ 1 (X ≥ 1)

Доступность: высокая

Надёжность: высокая



Подход баз данных: кворум

Кворумная запись защищает от потерь данных
Кворум гарантирует консистентность X → 1 (X ≥ 1)
Надёжно, но медленно



Кворумный кластер очереди

P L F F F F C C

Гарантии: X ≈ 1 (X ≥ 1)

Надёжность: высокая



Кворумный кластер очереди

Partition

Доступность: ограничена

Гарантии: X ≈ 1 (X ≥ 1)

Надёжность: высокая



Кворумный кластер очереди

P

F

F

L

C

C

Partition

Гарантии: X ≈ 1 (X ≥ 1)

Доступность: ограничена

Надёжность: высокая



Кворумный кластер очереди

Масштабируемость: нет

Гарантии: $X \approx 1$ ($X \geq 1$)

Доступность: ограничена

Надёжность: высокая



Реплицированные очереди, 1 из N

P

1 of N

C

C

Масштабируемость: да

Гарантии: X ≈ 1 (X ≥ 1)

Доступность: высокая

Надёжность: высокая



Кворумные очереди, 1 из N

Масштабируемость: да

Гарантии: X ≈ 1 (X ≥ 1)

Доступность: высокая

Надёжность: высокая



Кворумные очереди, 1 из N

Масштабируемость: да

Гарантии: X ≈ 1 (X ≥ 1)

Доступность: высокая

Надёжность: высокая



Кворумные очереди, 1 из N

P
P
1 of N
AZ1
AZ2
C
C

Масштабируемость: да

Гарантии: X ≈ 1 (X ≥ 1)

Доступность: высокая

Надёжность: высокая



Протоколы очередей и ограничения

Состояние задачи в соединении

Без состояния (HTTP/REST/SQS)



Протоколы очередей и ограничения

Состояние задачи в соединении

- Низкая задержка
- Мгновенный возврат
- Сложно масштабировать
- Жизненный цикл

Без состояния (HTTP/REST/SQS)



Протоколы очередей и ограничения

Состояние задачи в соединении

- Низкая задержка
- Мгновенный возврат
- Сложно масштабировать
- Жизненный цикл

Без состояния (HTTP/REST/SQS)

- Масштабирование
- HTTP-балансировка
- Нужен автовозврат



Мониторинг и эксплуатация

- Размеры очереди
  - Очередь всегда ограничена



Мониторинг и эксплуатация

- Размеры очереди
  - Очередь всегда ограничена
- Время
  - Полная обработка сообщения (QoS)
  - Время исполнения



Мониторинг и эксплуатация

- Размеры очереди
  - Очередь всегда ограничена
- Время
  - Полная обработка сообщения (QoS)
  - Время исполнения
- Количество повторов и потерь/отказов



Мониторинг и эксплуатация

- Размеры очереди
  - Очередь всегда ограничена
- Время
  - Полная обработка сообщения (QoS)
  - Время исполнения
- Количество повторов и потерь/отказов
- Поток сообщений



Мониторинг и эксплуатация

- Размеры очереди
  - Очередь всегда ограничена
- Время
  - Полная обработка сообщения (QoS)
  - Время исполнения
- Количество повторов и потерь/отказов
- Поток сообщений

- Логируйте сообщения!



Эксплуатация: планируйте отказ

- Настраивайте политики отказа
    - Перестаньте принимать новое
    - Уничтожьте старое
    - «Спасайте» выживших



Эксплуатация: планируйте отказ

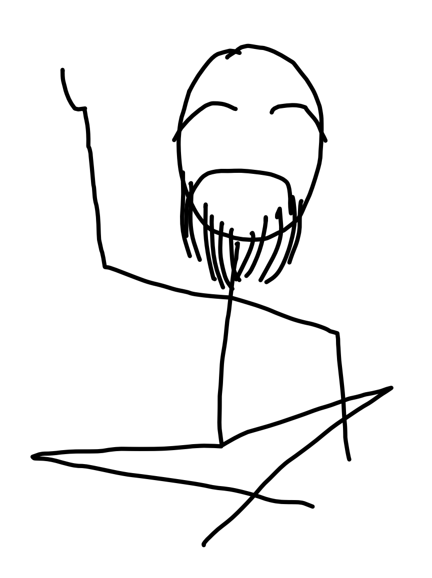
- Настраивайте политики отказа
  - Перестаньте принимать новое
  - Уничтожьте старое
  - «Спасайте» выживших
- Запланируйте падение
  - Для того, чтобы подняться



Эксплуатация: планируйте отказ

- Настраивайте политики отказа
  - Перестаньте принимать новое
  - Уничтожьте старое
  - «Спасайте» выживших
- Запланируйте падение
  - Для того, чтобы подняться

Не тот велик, кто
никогда не падал,
а тот велик — кто падал и
вставал!





Что же взять?

- Толерантность сервиса к потерям
- Организация передачи сообщений
- Высокая пропускная способность, масштабируемость



Что же взять?

- Толерантность сервиса к потерям
- Организация передачи сообщений
- Высокая пропускная способность, масштабируемость

- **NATS**
- NSQ
- ZeroMQ



Что же взять?

- Быстро попробовать
- Соединить микросервисы или k8s
- Сервис работает в облаке



Что же взять?

- Быстро попробовать
- Соединить микросервисы или k8s
- Сервис работает в облаке

- **SQS** (Simple Queue Service)
  - Amazon, Mail.ru Cloud, Yandex, …
- CloudAMQP
- Простые брокеры: RabbitMQ, NATS (следите за надёжностью)



Что же взять?

- Организовать стриминговую архитектуру
- Нужна высокая сохранность
- Требуется строгий FIFO



Что же взять?

- Организовать стриминговую архитектуру
- Нужна высокая сохранность
- Требуется строгий FIFO

- Apache **Kafka**
- NATS JetStream
- Tarantool Enterprise



Что же взять?

- Сложные сценарии очередей
- Отложенные задачи, перепостановка
- Произвольные топологии, собственные алгоритмы
- Зависимые сценарии



Что же взять?

- Сложные сценарии очередей
- Отложенные задачи, перепостановка
- Произвольные топологии, собственные алгоритмы
- Зависимые сценарии

- **RabbitMQ**
- Tarantool Queue / **Tarantool**



Спасибо за внимание!



Мои контакты:
Email: mons@cpan.org
Telegram: @inthrax
@tarantool_ru

